Universidad Fidélitas **I Cuatrimestre, 2014**
Programación III **Valor 30%**
Lic. Randy Valverde Valverde **14 de febrero, 2014**

El presente proyecto se deberá entregar el día **25 de abril, 2013**, **durante la primera hora de clase**. **Después de este tiempo no se recibirá ningún trabajo**. El proyecto debe ser elaborado en forma individual o en parejas. Si entre personas del mismo horario, existen dos o más trabajos iguales, serán anulados y será causal para la pérdida del curso, sin derecho a reclamos. De igual manera aplicará si se comprueba que el trabajo ha sido realizado por terceras personas o si existe copia parcial o total de códigos de Internet. **El día de la revisión del proyecto se realizará una comprobación de cada proyecto entregado**.

Generalidades de la Aplicación

- La aplicación a desarrollar, debe de ser una amigable para el usuario. La interfaz que desarrolle debe de ser clara, concisa y de fácil uso para el usuario de la aplicación.

- La aplicación debe contener todas las validaciones necesarias, que le indiquen al usuario que no ha ingresado información en campos requeridos o que ingrese datos en el formato adecuado. Tome cuenta formatos para fechas, valores monetarios, etc.

- Utilicé iconos o imágenes representativas, esto con el fin de guiar al usuario y que la imagen o icono que puso en la aplicación realice la acción o proceso al cual está representando.

- Desarrolle un estándar en su codificación. Asigne nombres claros a las variables con el fin de un adecuado mantenimiento al código de la aplicación. Realice comentarios en el código donde considere necesario.

- Recuerde que hay usuarios que no usan el Mouse para pasarse de campo, sino que usan la tecla TAB, para ello genere una adecuada navegación entre los campos de los formularios.

- Utilice accesos directos en sus menús (ejemplo Ctrl + A = Ayuda).

- Utilice todos los recursos que brinde la herramienta para el desarrollo del sistema y que son autorizados en el curso

- Recuerde utilizar la propiedad MaxLength en las cajas de textos, para limitar la entrada de datos al usuario o la cantidad de caracteres que puede ingresar en cada campo.

- Es de vital importancia comprender y aplicar la función de aleatoriedad para la ubicación de los tiburones en cada momento que se inicie el juego

Herramienta de Desarrollo

- Visual Basic. Net

Motor de base de datos

- SQL SERVER 2008.

Aspectos Administrativos

- El proyecto puede ser elaborado en forma individual o en parejas. Los equipos de trabajo son invariables hasta la entrega del proyecto. Si entre dos parejas del mismo horario o en distintos horarios existen dos o más trabajos iguales, serán anulados y será causal para la pérdida del curso, sin derecho a reclamos. De igual manera aplicará si se comprueba que el trabajo ha sido realizado por terceras personas o si existe copia parcial o total de códigos de Internet.

- El día de la entrega del proyecto se realizará una comprobación a cada grupo de trabajo, con el fin de verificar la veracidad del proyecto.

  La comprobación del proyecto se realizará con cada estudiante de cada grupo de trabajo, por lo que cada alumno debe demostrar la correcta funcionalidad del mismo y poder contestar cualquier consulta sobre la forma en la que se haya desarrollado cualquiera de los elementos de su proyecto.

- Debe cumplir con la integridad referencial en la base de datos.

- Debe desarrollar todo elemento indicado en el presente documento. No desarrolle lo que no se solicita, pues esto no será tomado en cuenta para su nota.

Descripción general

Se le ha contratado a usted como desarrollador para programar el juego Busca tesoros siguiendo la misma lógica del conocido juego de Windows Buscaminas. Adjunto una descripción lógica del juego para que le sirva como guía: http://es.wikipedia.org/wiki/Buscaminas

El juego consiste en despejar todas las casillas de una pantalla que no oculten a los tiburones.

Algunas casillas tienen un número, este número indica la cantidad de tiburones que existirán en el tablero de juegos, las mismas son todos aquellos peligros que suman todas las casillas circundantes. Así si una casilla tiene el número 3 significa que de las ocho casillas que hay alrededor (si no está en una esquina o borde) hay 3 con tiburones y 5 libres. Si se descubre una casilla sin número indica que ninguna de las casillas vecinas tiene tiburones y estas se descubren automáticamente.

Si se descubre una casilla con un tiburón se pierde la partida.

La forma de ganar es sencilla, una vez que se ha descubierto todo el tablero se le dirá que ha sido el ganador del tesoro y procederá a enviar las estadísticas (tiempo de duración de la partida, cantidad de vidas perdidas) al correo electrónico del jugador.

Consecuentemente esto implica que el usuario para poder ingresar a jugar el juego debe de indicar al menos el correo electrónico y el año de nacimiento.

El año de nacimiento no servirá para determinar alguno de los 2 niveles de dificultad (puede ser configurable a mas niveles de dificultad), ya sea avanzado (para mayores de 15 años, tablero de 16x16 casillas 40 tiburones), principiantes (para menores de 15 años con un tablero de 8x8 con 10 tiburones).

Una parte importante del proyecto es guardar los registros de los usuarios y las puntuaciones obtenidas, las cuales van a estar relacionadas con la cantidad de vidas utilizadas y el tiempo requerido.

Descripción de Entidades

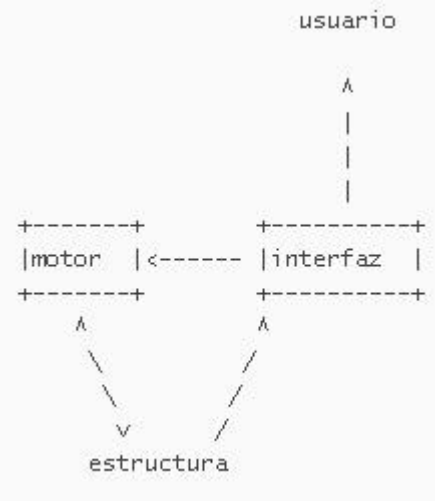
Entidad	Descripción
Parámetros	Permite configurar el funcionamiento de la aplicación Par_Juego: determina la cantidad de niveles que va a tener y las edades para ca uno de los niveles asi como los tableros y cantidad de tiburones
Jugadores	Catálogo de usuarios que han jugado y van a jugar el juego
Administradores	Estructura del administrador del juego
Puntuaciones	Puntuaciones del juego, por tiempo, vidas.
Bitácora	Bitácora de los administradores y las veces en que ha actualizado su contraseña así como los cambio o

Entidad	Descripción
	configuraciones echar para el sistema.

Modelo de datos

Debe de ser desarrollado por usted

El sistema que implementa el juego se construirá de acuerdo con la siguiente arquitectura:



Este es un modelo simple donde tenemos dos capas o módulos: la interfaz y el motor.

La interfaz se encarga de realizar el diálogo con el usuario (el jugador), capturar sus entradas e invocar las operaciones asociadas. La interfaz también maneja la imagen del juego en pantalla actualizándola cada vez que hay cambios. En resumen, se encarga de todo lo que tenga que ver con entrada y salida. La interfaz no realiza ninguna modificación directa sobre la estructura de datos.

El motor es el módulo que trata con la estructura de datos que representa el juego en un estado determinado, recordemos que este motor será el encargado de aplicar la aleatoriedad al juegos además de servir de conexión entre los datos solicitados y la base de datos. Este módulo estará compuesto por todos los subprogramas necesarios para ejecutar las acciones del jugador y reflejar estas en la estructura de datos. El motor no realiza ninguna operación de entrada y salida.

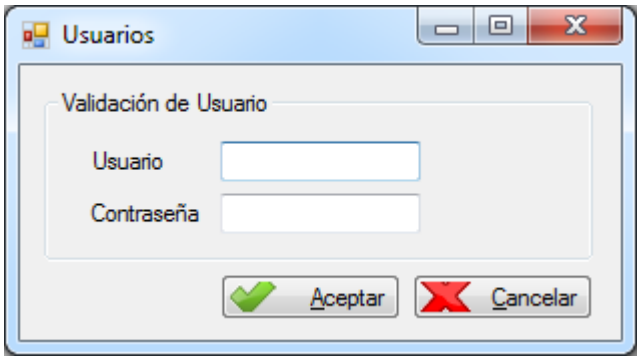
Estructura del menú principal

Sistema Administrador de Juego [Parametros.Juego]			
Sistema	Niveles	Consultas	Ayuda
Parámetros	Puntuaciones Generales	Navegador del juego	Acerca de
Registro	Punt. Nivel 1	Historia del juego	
----------	Punt. Nivel 2	Logros	
Salir	Punt. Nivel X		

La barra de herramientas debe incluir al menos las funcionalidades:

- Parámetros
- Niveles
- Ayuda

Ejemplo de la ventana de autenticación del administrador



Para la ventana de parámetros solo podrá ingresar el administrador del sistema con la contraseña (juego@123-) y la podrá cambiar dentro de este módulo.

El resto de pantallas de interfaces será libre a creatividad del usuario siempre y cuando cumpla con los requerimientos antes solicitados

Aspectos a Evaluar

Requerimiento	Valor	Puntaje
Base de Datos	5 pts.	
Validaciones	5 pts.	
Pantallas	10 pts.	
Autenticación del usuario administrador	2 pts.	
Mantenimiento	2 pts.	
Parámetros de configuración	2 pts.	
Juegos	2 pts.	
Niveles	2 pts.	
Juego	2 pts.	

Cronometro	2 pts.	
Cantidad de intentos	2 pts.	
Tablero de juego (función aleatorio)	10 pts.	
Terminación del juego y envió del correo	4 pts.	
Consultas	2 pts.	
Navegador del juego	2 pts.	
Historia del juego	2 pts.	
Logros	2 pts.	
Acerca de	2 pts.	
Consideraciones importantes	10 pts.	
Integridad referencial	10 pts.	
Estética del juego	10 pts.	
Amigabilidad de la interfaz	10 pts.	

Considerar la siguiente guía lógica (de seudocódigo) para la elaboración de los juegos:

La estructura

La estructura de datos que representa el juego de buscaminas es la siguiente:

Código:

```
type
  TipoEstadoJuego = (jugando,ganado,perdido);
  TipoJuego = record
      estado       : TipoEstadoJuego;
      tablero      : TipoTablero;
      tiburones,                  (* cantidad de tiburones en el tablero *)
      marcadas,                   (* cantidad de celdas marcadas *)
      descubiertas : integer;     (* cantidad de celdas descubiertas *)
  end;
```

El juego es un registro con los siguientes campos:

- estado: Contiene el estado del juego en un momento dado.
- tablero: el tablero con las celdas
- tiburones: cantidad de tiburones en total en el tablero
- marcados: cantidad de tiburones ya marcadas por el usuario
- descubiertas: cantidad de celdas que están descubiertas

El tablero

El tablero se representa con la siguiente estructura:

Código:

```
RangoFila    = 1..MAX_FILAS;
   RangoColumna = 1..MAX_COLUMNAS;
   TipoTablero = record
      celdas  : array[RangoFila,RangoColumna] of TipoCelda;
```

```
      topeFila    : RangoFila;
      topeColumna : RangoColumna;
   end;
```

Esta estructura es una matriz (ver Arreglos Bidimensionales) con dos topes. De esta manera se pueden representar tableros de diferentes tamaños. Las celdas válidas de la matriz son aquellas cuyas coordenadas (i,j) son tales:

• 1<= i <= topeFila
• 1<= j <= topeColumna

Las constantes **MAX_FILAS y MAX_COLUMNAS** se suponen definidas con valores apropiados. Estos valores están definidos en la interfaz, de manera que el estudiante no necesita conocerlos.

La celda

La celda se representa con la siguiente estructura:

Código:

```
TipoEstadoCelda = (oculta,marcada,descubierta);
   TipoCelda = record
      estado : TipoEstadoCelda;
      case tieneTiburon :Boolean of
         True  : ();
         False : (TiburonCircundantes :integer)
      end;
```

La información que tiene una celda es la siguiente:

- estado : indica si la celda fue marcada, descubierta o continúa
- oculta (cuando decimos oculta nos referimos a oculta y sin marca).
- tieneTiburon: es un booleano que vale true si y sólo si la celda contiene una mina.
- TiburonCircundantes: indica la cantidad de tiburones que hay en las celdas circundantes. Este valor puede ser 0.

Considere que durante el desarrollo del proceso se estarán realizando avances en lógicos que le servirán de guían en el sitio web http://www.randyvv.com

Evaluación

Parte escrita (Valor 50 pts.)		Parte programada (Valor 50 pts.)	
Manual de usuario	20 pts.	Funcionalidad	25 pts
Manual del programador	20 pts.	Cumplimiento	25 pts.
Portada	2 pts.		
Introducción	3 pts.		
Conclusión	3 pts.		

Guía para elaboración de los proyectos programados

El proyecto programado es una actividad académica que persigue ampliar los conocimientos del estudiante, a partir de la base didáctica impartida en clase por el profesor. Por lo anterior, es que el cronograma de los proyectos se ajusta al orden de aprendizaje logrado por el estudiante durante el curso lectivo, situación que además, le permite avanzar en el desarrollo del proyecto en forma paulatina, sin generarle desgaste excesivo.

Con esta obligación académica se pretende que el estudiante logre un dominio suficiente de la herramienta de programación, por lo que, es viable que exista algún nivel de investigación referente al tema, que permita mejorar el producto final solicitado por el profesor. Se sugiere que para la elaboración de un proyecto programado el estudiante realice, las actividades que se detallan a continuación:

I. Análisis e investigación

En esta etapa de aproximadamente una semana, el estudiante debe delimitar el problema, para lo cual es indispensable el estudio del enunciado principal, la conversación con otros compañeros y las consultas extra clase al profesor.

II. Planeamiento y diseño

Una vez delimitado el problema y hecha la proyección de las alternativas de solución existentes, se debe destinar tiempo suficiente para planear y diseñar visualmente el producto final, asimismo, se debe establecer formas de navegación dentro del programa, así como la interfaz que debe tener la herramienta con el usuario final.

III. Desarrollo

Esta es la etapa más larga del proceso, dado que requiere la dedicación profunda de parte del estudiante, que le permita resolver con la herramienta de software indicada, el problema existente.

IV. Pruebas y documentación

La etapa final del proceso se completa con una serie de pruebas de ejecución, que han de permitir medir y conocer el grado de satisfacción del cliente hacia el producto que se ha de entregar, incluida la documentación del mismo.

Es imprescindible que los estudiantes se integren al producto y a los resultados que se esperan obtener, dado que la evaluación integra el proceso completo, la misma se describe a continuación:

Producto		**Avances**	**Doc.Final**	**Progr. Final**
Avance # 1 (I etapa) Semana 8	5%	5%	50 pts.	50 pts.
Avance # 2 (II Etapa) Semana 12	10%			
Avance # 3 (100% del proyecto terminado) Semana 15	10%			
TOTAL GENERAL		**30%**		